verizon

White paper

Zero Trust Architecture

February 2021

Brent Bilger

Table of Contents

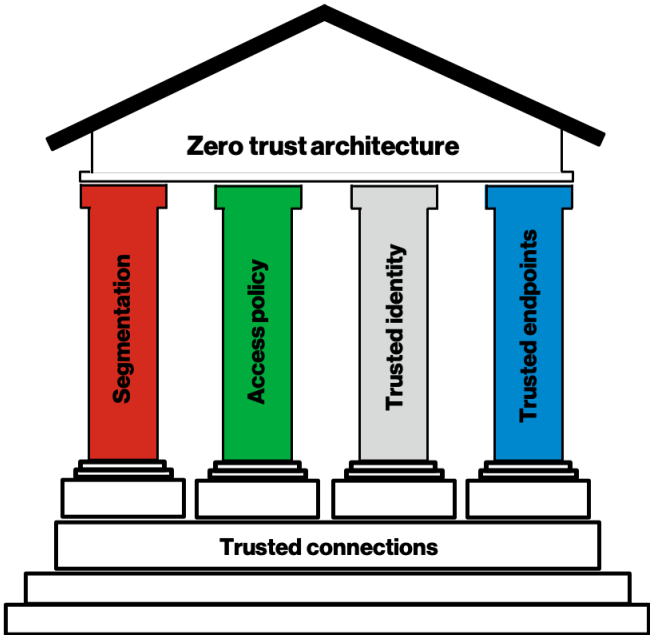
Executive summary

The National Institute of Standards and Technology (NIST) Special Publication (SP) 800-207 was issued in its final form on August 11, 2020. The document defines how the Department of Defense (DoD) and other government organizations should implement portions of their data network. Moreover, it puts to rest the multiple opinions of numerous vendors regarding what "The Zero Trust Architecture" actually is. The defined architecture recognizes the need to:

- isolate servers from the network to defeat exploitation
- provide multifactor authentication for access to all servers, and
- cryptographically secure client-server connections to defeat man-in-the-middle attacks

Throughout the SP 800-207 document, the **Software Defined Perimeter** as defined by the Cloud Security Alliance is referenced as a preferred implementation of the Zero Trust Architecture. The Software Defined Perimeter isolates servers from the network to defeat server exploitation, implements multifactor authentication to defeat credential theft, and provides impregnable tunnels to defeat man-in-the-middle attacks – meeting the objectives of the Zero Trust Architecture. The SP 800-207 document also goes on to define how other security controls can be used with the Software Defined Perimeter to increase the trust level of users and endpoints.

This paper clarifies the key components of the Zero Trust Architecture in terms that are familiar to security professionals. It defines the four pillars of the new standard: segmentation, access policy, trusted identity, and trusted endpoints, and explains how they can be implemented such that only trusted users on trusted devices can access trusted applications over trusted connections – the nirvana of secure networking.



Why do we need the Zero Trust Architecture?

Despite the plethora of cybersecurity tools, data breaches and compromised servers are at an all-time high. A few sobering statistics:

- **96% of the time, pen testers found at least one in-production vulnerability**. In Rapid7's 2018 "Under the Hoodie" survey looking at 200+ pen tests on enterprise networks, it was found that if the pen testers got access to the internal network, 96% of the time they were able to compromise one or more servers based on server software vulnerabilities.
- **96% of the time pen testers found at least one network or service misconfiguration**. The same survey found that the pen testers had similar levels of success in compromising network infrastructure components (e.g., switches, router, firewalls, etc.) or an application on a server because of misconfigurations (such as a default password on the device).
- **80% of hacking-related breaches leveraged weak or stolen passwords**. According to the 2020 Verizon Data Breach Investigations Report 4 out of 5 data breaches leveraged compromised passwords.
- **35% of exploitation activity involved man-in-the-middle attacks.** According to the IBM X-Force 2018 Threat Intelligence Index more than a third of attacks used some kind of man-in-the-middle attack.

The Zero Trust Architecture was developed to mitigate these attacks by using the following techniques:

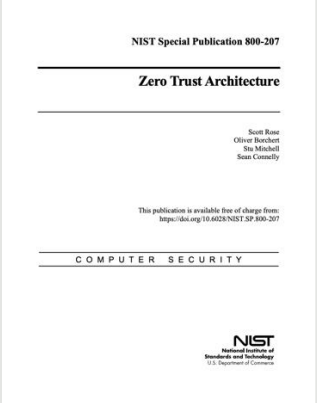
- Isolate servers to defeat exploitation of vulnerabilities and misconfigurations
- Apply multifactor authentication everywhere to defeat credential theft
- Create end-to-end impregnable encryption to defeat man-in-the-middle attacks

NIST SP 800-207

First released in draft form in 2019 and finalized in 2020, the NIST SP 800-207 document defines the Zero Trust Architecture. Some of the key takeaways are:

"Zero trust is a cybersecurity paradigm focused on... the premise that trust is never granted implicitly but must be continually evaluated."

"Zero trust architecture is an end-to-end approach to enterprise resource and data security... Traditionally, agencies (and enterprise networks in general) have [relied] on perimeter defense, and authenticated users are given authorized access to a broad collection of resources. As a result, unauthorized lateral movement within a network has been one of the biggest challenges for federal agencies."


NIST Special Publication 800-207

Zero Trust Architecture

Scott Rose
Oliver Borchert
Stu Mitchell
Sean Connelly

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-207

COMPUTER SECURITY

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Figure 1: cover of NIST SP 800-207

Implementing the Zero Trust Architecture

With the objectives in mind, how does one implement the Zero Trust Architecture? The answer lies in the four pillars of functionality that, once implemented, can help companies achieve a secure networking nirvana with the mantra "only trusted users on trusted devices can access trusted applications over trusted connections." Let's take a look at each of the pillars.

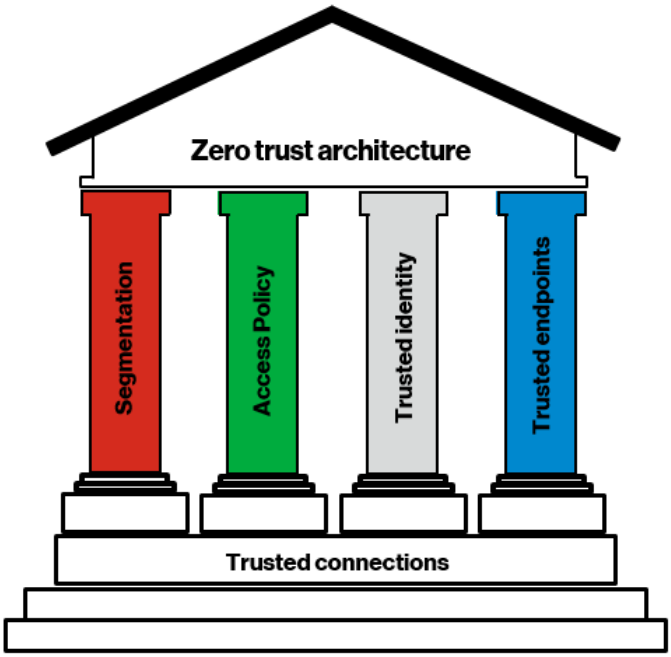


Figure 2. The Zero Trust Architecture consists of four pillars of functionality such that only trusted users on trusted devices can access trusted applications over trusted connections.

Segmentation

Segmentation is the primary control used by SP 800-207 to create the least privilege, need-to-know architecture. At a high level, there are three ways a network can be segmented: client-to-client, client-to-server, and server-to-server. While each type is important for different reasons, **the NIST SP 800-207 Zero Trust Architecture only focuses on client-to-server segmentation to implement zero trust access**.

Figure 1 of the SP 800-207 Zero Trust Architecture document is reflected below (Figure 3). On the left we see the environment of the users and that it is untrusted. On the right, we see the environment of the enterprise applications and that it is trusted. In between, the Policy Decision Point and the Policy Enforcement Point of the Zero Trust Architecture isolate the untrusted resources on the left from the trusted ones on the right. Basically, isolating the users and their devices on the left from the enterprise applications on the right.

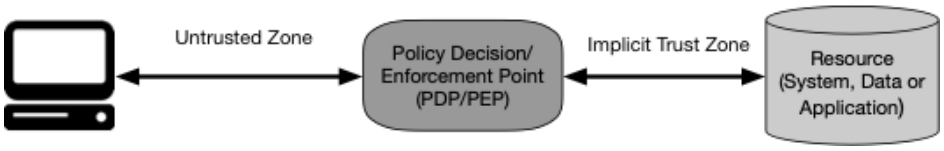


Figure 3: The first figure in NIST SP 800-207 where the Zero Trust Architecture isolates the untrusted user environment on the left from the trusted environment on the right.

If the untrusted users and untrusted endpoints on the left ***are isolated from*** the servers on the right, then endpoints that have been compromised by ***adversaries can be segmented from the business-critical data*** in the servers on the right.

According to section 3.1.3 of the NIST SP 800-207 Zero Trust Architecture, a preferred way to implement client-to-server segmentation is via the Software Defined Perimeter (SDP) defined by the Cloud Security Alliance. Figure 4 shows a common depiction of SDP with the users on the left in the untrusted zone and the applications on the right in the trusted zone. It should be immediately obvious why the SP 800-207 document calls out the Software Defined Perimeter for implementing the Zero Trust Architecture. The Software Defined Perimeter implements the Policy Decision Point and Policy Enforcement Point of the Zero Trust Architecture.

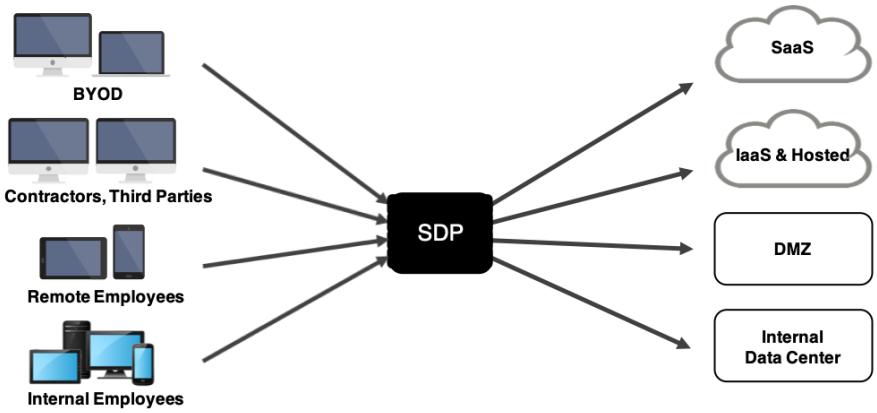


Figure 4: A simplified version of the Software Defined Perimeter architecture showing the untrusted users on the left segmented from the trusted applications on the right by the Software Defined Perimeter.

Access Policy

Now that the servers, applications, data, and infrastructure components have been isolated from the untrusted environment, we need a way for authorized, trusted users on authorized, trusted devices to access the isolated applications. The SP 800-207 document is explicit on the use of least privilege, whereby only users who "need-to-know" are given access to the isolated applications.

Primary Access Policy

The primary access policy is implemented by creating groups in the Enterprise Directory (typically, Microsoft Active Directory), where users that are members of a group are entitled to access one or more isolated applications. **In practice, "users being members of a group in the Enterprise Directory" is the primary access policy to obtain least privilege access.** Therefore, the third pillar of the Zero Trust Architecture, "Trusted Identity" becomes very important because it enforces that the user ***is*** who the user ***claims to be***.

Secondary Access Policies

While the primary access policy is binary (the user is either in a required group in the enterprise directory or not, and, therefore, is either entitled to access the isolated application or not) the secondary access policies are less straightforward in their applicability to all isolated applications. The secondary access policy is about determining whether or not the user's device can be trusted – that is, whether or not the device may have been compromised by an adversary. With the trust level of the PC determined, the Policy Decision Point can then apply a policy as to whether or not the device should be allowed to access some or all isolated applications to which the user is authorized. As we will see, trust of the endpoints is determined by some traditional security tools, but in a way that enables the Policy Decision Point to use their knowledge of the trust of the endpoint to respond by either allowing, or not allowing, access to isolated applications.

Tertiary Access Policies

There can be tertiary access policies, too. These would include at what day-of-week or time-of-day the user is trying to access the protected applications. It may not be appropriate for some users to access the data during certain days of the week or periods of the day. Similarly, country of origin may be a factor. There may be some countries in the world where the enterprise would never allow access to selected, isolated applications. Another factor that may be applied is that the traffic may have to originate within the internal network (i.e., remote access to the application is not allowed), or may even require the user be in a "secure facility" where entry to the facility is highly regulated. The list is almost endless but tertiary access policies mostly conform to time and location.

Summary of the Access Policies

In summary, if the user is authorized as defined by membership in a group in the enterprise directory (primary policy), and the user's device is trusted to the degree necessary for a particular application (secondary policy) and the user is attempting to access the protected application from a permitted location during a permitted time (tertiary policy), then the Zero Trust Architecture provides access.

Trusted identity

If group membership in the Enterprise Directory is the primary access policy, then we must make sure that users ***are*** who they say ***they are*** and that the Enterprise Directory service has not been compromised. Basically, we need to know that the user's identity, and the representation of that identity, can be trusted.

Stolen and weak passwords are a key factor in many data breaches. Therefore, the Zero Trust Architecture must make sure that ***the user is who they claim to be***. The primary method of making this happen is via multifactor authentication (a.k.a., 2-factor authentication). And the multifactor authentication needs to defeat both compromised passwords and compromised representations of the passwords such as NTLM hashes and Kerberos tickets (i.e., the multifactor authentication needs to defeat pass-the-hash and pass-the-ticket attacks). While typical multifactor authentication products can defeat password compromise, those same multifactor authentication products may not be able to defeat pass-the-hash or pass-the-ticket attacks. ***Specifically, if the multifactor authentication is only applied during user authentication, then it may not be able to defeat those attacks***.

Finally, "the groups the user is a member of" must be securely transmitted from the Enterprise Directory to the Policy Decision Point such that the latter can determine whether or not to give the user access to the enterprise applications.

In summary, we need to isolate the Enterprise Directory from adversaries, make users apply multifactor authentication to gain access to reading the directory, and transfer "the groups the user is a member of" to the Policy Decision Point via some secure, encrypted method. Basically, ***we need to implement the Zero Trust Architecture to access to the Enterprise Directory to ensure trusted identity***.

Trusted endpoints

As mentioned above, the primary access policy is based on the user being a member of a group in the Enterprise Directory and the secondary policies are about determining whether or not the client endpoints can be trusted.

A good way to analyze trust is via the NIST cybersecurity framework, consisting of five security-related functions: identify, protect, detect, respond, and recover. However, relative to the Zero Trust Architecture the Policy Decision Point needs to know that the "protect" controls are in place and the "detect" controls believe the endpoints are trustworthy. Based on those two pieces of information, the Policy Decision Point will "respond" by either allowing the connection or blocking it. To help analyze "protect" and "detect" controls, each can be further divided into two groups: endpoint-deployed and network-deployed.

Endpoint-deployed detect controls

Endpoint-deployed "detect" controls include those controls listed in Figure 5. It's rare that enterprises employ all of these controls, but as more endpoint protection products incorporate more of these functions into a single product, more of these controls are getting deployed. The trust assessment of each of these controls is fed into the Policy Decision Point such that it can make a policy decision based on the trust of the endpoint.

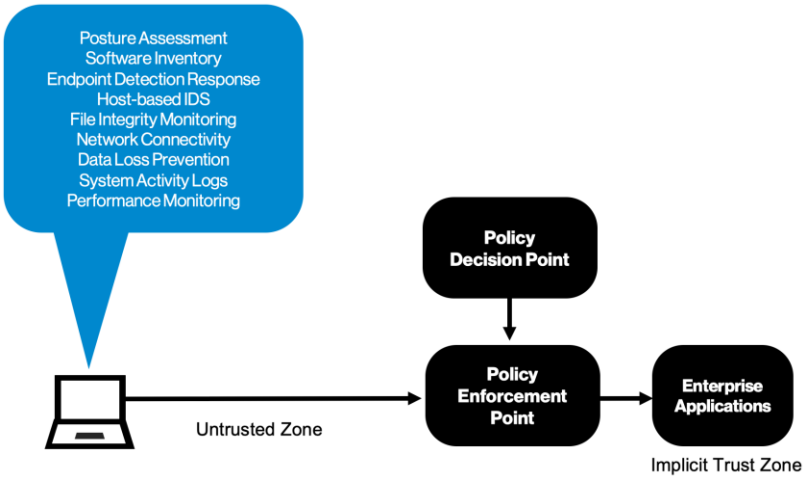


Figure 5: Endpoint-deployed "detect" controls send their trust assessment to the Policy Decision Point

Endpoint-deployed protect controls

The purpose of "posture assessment" is to monitor the "protect" controls that need to be in place. Figure 6 shows the types of protect controls that can be detected by posture assessment. Having all, or some subset, of these controls in place is also a common policy for the Policy Decision Point.

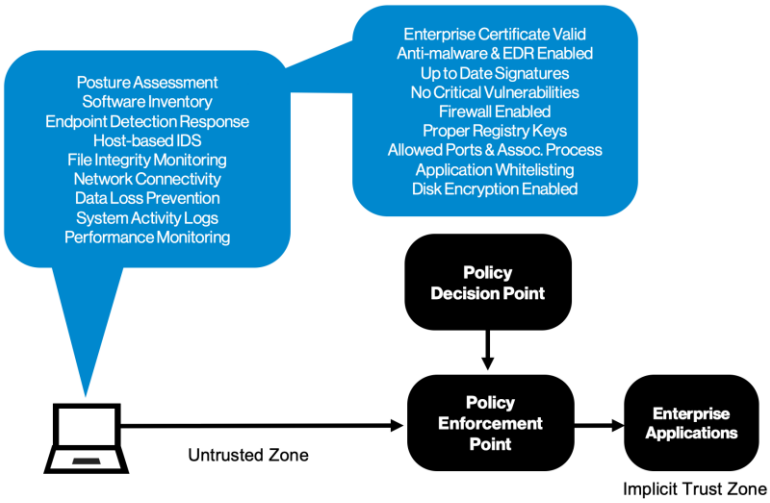


Figure 6: Posture assessment reports the presence and operational readiness of the "protect" controls to the Policy Decision Point.

Network-deployed detect controls

The possible network-deployed "detect" controls include those listed in Figure 7. These controls are in the unique position of not only reporting on the specific endpoint in question, but on all the neighbors of that endpoint. These controls are also able to use threat intelligence from around the world to provide additional input to the Policy Decision Point.

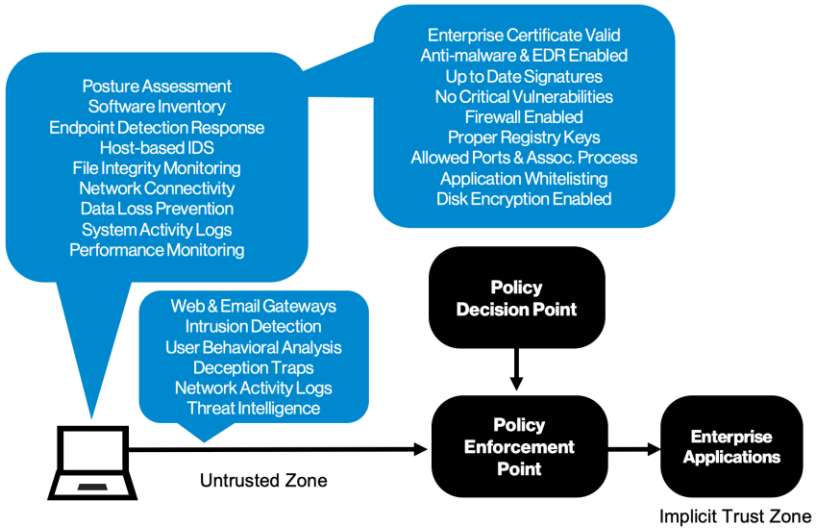


Figure 7: Network-deployed "detect" control report on the network activity of an endpoint and on all the neighbors of the endpoint.

Network-deployed protect controls

Interestingly, many network-deployed "protect controls" also act as "detect controls". Web and email gateways have alerting mechanisms when malware is detected, Intrusion Prevention Systems (IPS) have the same alerts as Intrusion Detection Systems (IDS), Web Access Firewalls (WAF) alert on suspicious traffic, Security Orchestration, Automation and Response (SOAR) systems often require user confirmation before responding, firewalls produce logs of blocked traffic, and so on. Essentially, these protection controls are blocking the endpoint and alerting on the IP address of the endpoint that gets blocked.

In summary, the intelligence of a number of "protect" and "detect" controls are aggregated by the Policy Decision Point to make the decision of how "trusted" the user's device is, and which applications the user's device is authorized to access.

Over trusted connections

If the primary policy confirms that the user is authorized to access one or more isolated applications, and if the secondary policies confirm that the user's endpoint is trusted to the degree necessary for the applications, and if any tertiary policies are met, then an encrypted tunnel is created from the user's device to the Policy Enforcement Point. All communication between the applications on the user's device and the isolated applications are carried over this encrypted tunnel. That means that

unencrypted protocols such as HTTP and Telnet become encrypted, and ***encrypted*** protocols such as HTTPS and SSH get double encrypted where the second layer of encryption encrypts all cleartext components of the HTTPS and SSH handshakes. This second layer of encryption is also going to defeat man-in-the-middle attacks (see the "How to Defeat Man-in-the-Middle Attacks" white paper).

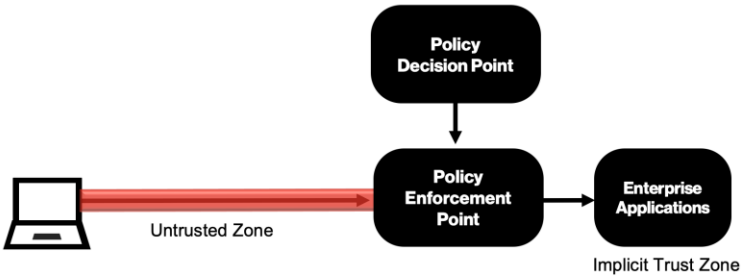


Figure 8: After all the policies are applied, an impregnable tunnel is created between the client and the Policy Enforcement Point. This completes the nirvana of secure networking: only trusted users on trusted devices can access trusted applications ***over trusted connections****.*

Therefore, the question becomes, "How does one create a trusted connection?" First, the protocol being used must ensure that the initial connection cannot be compromised by adversaries. Second, both sides of the connection must be mutually authenticated to the other side with digital certificates that cannot be cryptographically compromised. Third, the protocol being used must ensure that the algorithm of TLS being used cannot be downgraded or modified in any way. Fourth, the digital certificates used in the TLS handshake cannot be forged. And fifth, DNS poisoning must be defeated.

Single Packet Authorization

Over the years, there have been many types of attacks on TLS. One of these is a denial of service attack on the server based on the asymmetric cryptographic workload of setting up the TLS connection. To defeat that attack, some implementations of the Software Defined Perimeter have implemented the Single Packet Authorization (SPA) protocol. This protocol occurs before TLS and uses a cryptographic shared secret between the endpoint and the Policy Decision Point to generate a one-time password. Since the Single Packet Authorization protocol is cryptographically efficient to evaluate, it can defeat the denial of service attack.

Mutual TLS

Mutual TLS should be used to generate the encrypted tunnel. Regular TLS only authenticates the server to the client, but mutual TLS also authenticates the client to the server at the same time. Alternatives to mutual TLS are to either have the user log into the Policy Enforcement Point with a username and password, or use some token that represents the user. However, the former is susceptible to password theft and the latter to token theft (e.g., pass-the-hash).

Unalterable TLS Suite

TLS has multiple versions (suites), each with multiple algorithms, which can be used. During the TLS handshake the client states the TLS versions it supports, and the server picks the most secure one. However, a man-in-the-middle can intercept the TLS handshake and alter the supported versions of the client such that the underlying cryptography of the resulting version can be compromised. Therefore, the Zero Trust Architecture must control both sides of every connection so that only one TLS version is used – one that cannot be downgraded or modified.

Pinned Certificates

Another optional configuration of the Software Defined Perimeter is pinned certificates. Browsers trust hundreds of Public Key Infrastructure (PKI) Certificate Authorities (CA). Therefore, it is possible for an adversary to obtain a forged certificate of a server in order to impersonate the Policy Decision Point. However, if the Software Defined Perimeter uses pinned certificates, then it does not put its trust in public Certificate Authorities – only in the ones dedicated to the Software Defined Perimeter. This can defeat man-in-the-middle attacks based on forged certificates.

Certificates with a Common Name that is an IP address

The great majority of PKI certificates of servers on the Internet use the server's host name for the Common Name (CN) of the certificate. For the client to authenticate the server, it compares the "hostname it is attempting to access" to the "hostname on the certificate" AND the "IP address it receives from DNS for that hostname" with the "IP address it receives from the server." If both comparisons match, then the server is authenticated. Unfortunately, this means that if an adversary can "poison the DNS" then the adversary can become a man-in-the-middle. However, if the server (or in this case the Policy Decision Point and Policy Enforcement Point) uses an IP address for the Common Name on each certificate instead of the hostname, the poisoning of DNS can be defeated.

Summary of a Trusted Connection

With the above protocols and algorithms in place, we can ensure that adversaries cannot defeat the secrecy or integrity of communication between the user's device and either the Policy Decision Point or the Policy Enforcement Point.

Use cases

The individual use cases for a Zero Trust Architecture fall into four areas: 1) remote access, 2) internal access, 3) SaaS, and 4) IoT.

Zero Trust VPN

When COVID-19 struck, companies were generally more concerned with expanding their existing remote access than securing it. But companies now face a future where remote access is an ongoing business requirement and they are questioning whether remote users need to have full access to the internal network.

Remote access for third parties and employees

The Zero Trust Architecture can replace the traditional remote access VPN of users coming from the Internet to access data center and cloud applications. The key is that users get access to their authorized applications and nothing else. Also note that this use case is for both employees and third parties. The latter should only be able to access specific authorized applications – period – without visibility into the server, infrastructure or network. Once that mechanism is in place for third parties, it can also be leveraged for employees.

Access to multiple clouds and data centers

For users accessing multiple clouds and hosting centers simultaneously, Zero Trust Architecture connects to all those locations directly with no hairpinning of data. Therefore, it offers the users a lower latency and better performance than traditional remote access VPN solutions with only one entry point to the enterprise network and lots of hairpinning of data.

Internet-accessible Secure Enclave

"Secure Enclave" is an increasingly common term where an application is put in a Virtual Private Cloud (VPC) and accessed with the Zero Trust Architecture. The advantage is that the application is accessible from around the world but invisible to unauthorized users. Note that the Zero Trust Architecture is one of the few products that can create a secure enclave because it does multifactor authentication ***before*** user log in. Therefore, the servers in the secure enclave do not need to be exposed to the Internet to provde user login prior to multifactor authentication.

Zero Trust internal network

Many companies are starting their journey to the Zero Trust Architecture with a Zero Trust VPN, but significant value comes from continuing that architecture into the internal network where enterprise applications can be protected from adversaries inside the network.

Business-critical applications

Typically, there are few truly "business-critical" applications compared to the many applications that run inside a corporate network, but if an adversary compromises one or more of those business-critical applications, it can have serious consequences. Applications dealing with intellectual property, financial data, and personal employee information should be immediately put behind a Policy Decision Point Gateway, thus implementing the Zero Trust Architecture by isolating them from all internal users while providing access to authorized users on authorized devices.

Unsupported operating systems and applications

A common use case for the Zero Trust Architecture is applications on an older operating system that cannot be upgraded. There may be less than 100 users for each of these applications, but those users are often distributed throughout the company. Therefore, simple firewall rules do not work and exposing these servers to the internal network is a very bad idea. At the very least it will cause an audit finding, and if an adversary gets on the internal network it's likely that this server will be an early target. Put the server on its own VLAN and protect it with the Zero Trust Architecture.

Privileged access

Another common internal network use case is privileged access to servers via RDP and SSH. The Zero Trust Architecture Policy Decision Point is effectively a hardened, Bastian server-like jump box which isolates privileged access to only authorized users on authorized devices and requires multifactor authentication for all access. Furthermore, it provides visibility of which authorized users on which authorized devices accessed which servers on which ports, when they accessed them, from where, and how much data they downloaded and uploaded. It's not just for servers as the Zero Trust Architecture can also be used to manage infrastructure. The admins and application owners can also be more efficient because they can use the native tools they prefer such as BASH, FileZilla, puTTY, RDP, etc.

Zero Trust Software as a Service

The journey to the Zero Trust Architecture may begin with remote access and continue with isolating business-critical applications within the network. Next is extending it to enterprise SaaS applications. There are two different models here. Your business may offer "product-as-a-service" or (more commonly) you "consume" SaaS applications. In both cases, security is required.

Protecting a SaaS

If your business offers its product as-a-service, and you (or your customers) want a high degree of security, then the Zero Trust Architecture should be a component of the service offering. In many ways, this is like third-party access to your network, but instead of your internal network it's access to the SaaS applications. As described in 800-207, to truly implement the Zero Trust Architecture, your customers will have to install an agent on each of their devices. Therefore, this architecture is not for your average SaaS. This is for the SaaS where security is very important.

Protecting access to a SaaS

A more common use case is that your company would want to extend the Zero Trust Architecture to any SaaS application that your company consumes, and there are two ways to do this. If the SaaS supports IP whitelisting for access to the SaaS, then one can IP whitelist access to the Zero Trust Architecture Policy Decision Point. This inherently provides multifactor authentication to the SaaS and defeats many types of man-in-the-middle attacks. Alternatively, for every SaaS that supports SAML, one can route the SAML authentication to the internal network such that only authorized users on authorized devices with access to the internal network can get access to the SaaS. This is easy to implement and defeats adversaries on the Internet accessing SaaS applications.

Internet of Things

Finally, there is the Internet of Things (IoT). Here, *the device is the user*. That is, either the device is authorized to access a server or it is not – just like a user in the traditional Zero Trust Architecture – meaning the primary access policy is still a binary decision on whether or not the device should be allowed access. And, the secondary access policy is the same as we saw above, as well. That is, "has the device been compromised?" Therefore, the Zero Trust Architecture can still be applied to IoT devices. First, isolate the servers the IoT devices will access. Second, create an access policy based on the type of the device. Third, perform a trust assessment of the device. Fourth, create trusted connected from the authorized, trusted devices to the isolated servers.

Verizon Zero Trust Architecture

Verizon has a history with the Zero Trust Architecture that dates back over a decade. We can use that knowledge, together with our strong vendor partnerships, to provide our customers a customized, best-of-breed Zero Trust Architecture.

The Invention of the Software Defined Perimeter

Starting in 2008, Vidder, Inc. invented the Software Defined Perimeter technology to defeat the three adversarial techniques introduced at the beginning of this white paper. By 2013, they had coined the term “Software Defined Perimeter” and deployed their product to multiple customers. And in 2014, Vidder wrote the “Software Defined Perimeter Specification version 1.0” adopted and published by the Cloud Security Alliance. Today, we see that NIST SP 800-207 the Zero Trust Architecture document references the Software Defined Perimeter often as a preferred implementation.

In 2018, Verizon acquired Vidder whose core brain trust remains with Verizon and continues to develop, deploy, and support Software Defined Perimeter solutions.

Strong vendor partnerships

Verizon maintains strong partnerships with industry-leading security vendors that offer Zero Trust Architecture products. Let Verizon work with you on a Zero Trust solution that is right for your organization.

Summary

By implementing the four pillars of the Zero Trust Architecture, one can enable the nirvana of secure networking: “Only trusted users on trusted devices can access trusted applications over trusted connections”.